Introducing Common Resources v2 Engine

Last Modified on 08/31/2021 9:55 am EDT

By default, any new common resources (common resources) created across environments now utilize the common resource v2 engine.

This article explains some of the differences between the v1 and v2 engines, as well as how to specify which engine you want new common resources to use.

Why Common Resource v2 Matters

Common Resource v2 is the new backend for SAP Open Connectors common resources. Most of the enhancements are user-facing, and upgrading should have no impact on the existing common resource API calls in any of your code. However, we do recommend that you follow the Recommended Actions to preserve availability in the future.

Common Resource v1 is not supported anymore, meaning we will not be adding new enhancements or resolving new defects for common resource v1. If you experience a bug with v1 we suggest you test with v2; if the issue persists, please submit a defect to product for further investigation. New common resource features such as Automapping or common resource Sharing will only be supported for common resource v2 users, so users will need to upgrade to take advantage of the latest and greatest enhancements. You can contact Customer Support for migrating your common resource engine from v1 to v2 or vice versa.

Differences between v1 and v2

New Data Structure

For common resources created using the v1 engine, the `level` property exists in the root level of the common resource object. However, for common resources created using the v2 engine, the `level` property is nested in the `configuration` parameter. Additionally, you now have access to the unique `fields[*].id` identifier for each field in your object.

Example v1 Structure

In this v1 example, `level` is at the root level and on the last line of the object:

```
{
  "id": 12345,
  "objectName": "songs",
  "fields": [
    {
      "type": "string",
      "path": "artists[*].name",
      "displayName": "artists"
    },
    {
      "type": "string",
      "path": "songName",
      "displayName": ""
    }
  ],
  "level": "account"
}
```

Example v2 Structure

In this v2 example, `level` is nested under `fields`:

```
{
    "id": 12345,
    "objectName": "songs",
    "fields": [
      {
        "type": "string",
        "id": 11059,
        "path": "artists[*].name",
        "level": "account",
        "displayName": "artists"
      },
      {
        "type": "string",
        "id": 11060,
        "path": "songName",
        "level": "account",
        "displayName": ""
      }
}
```

New /VDRs APIs

The common resource v2 engine includes the new collection of `/VDRs` APIs. A Postman collection for the API is available here (same link as earlier in this article), and will soon also be available via the API Docs in the UI.

Determining your Current Engine

If you do not know which engine your account is utilizing for your newly built common resources, you can check either through the UI or an API call. In addition, the selected version is managed independently between staging and production environments; changing the selected version in one environment does not automatically carry over to others.

Checking the Engine via UI

1. Sign in to SAP Open Connectors and click API Docs in the top-right corner.

2. From the Platform API Documentation column on the left, click Accounts.
3. Under the Accounts APIs list, click `GET /accounts/me`.
4. Click Try It Out, and then click Execute.
5. In the response body, find the `vdrVersion` parameter. The value listed indicates the engine being used.

Checking the Engine via API

1. Make the following API call: `http://api.openconnectors.ext.hanatrial.ondemand.com/elements/api-v2//accounts/me`
2. In the response body, find the `vdrVersion` parameter. The value listed indicates the engine being used.

Backward Compatibility

When you upgrade to the v2 engine, all existing v1 APIs are backward compatible and no immediate code change is required to begin using and testing the v2 engine.

The v1 APIs are deprecated and planned for retirement in 2021, so while no immediate code change is required, we recommended that you begin planning an upgrade.

Recommended Actions

We suggest you run tests against your common resources in a staging environment prior to January 31, 2021.

If you experience any issues with your common resources while testing in staging, please reach out to support or your Customer Account Manager.

Transformations via API

This section talks about mapping fields for transformations using APIs.

1. Construct a JSON body as shown below. For descriptions of each parameter, see Transformation JSON Parameters. The below body will create a common resource with two fields, **firstName** and **lastName.**

```
{
 "fields": [
   {
     "type": "string",
     "path": "firstName",
     "displayName": ""
   }, {
     "type": "string",
     "path": "lastName",
     "displayName": ""

 ]
}
```

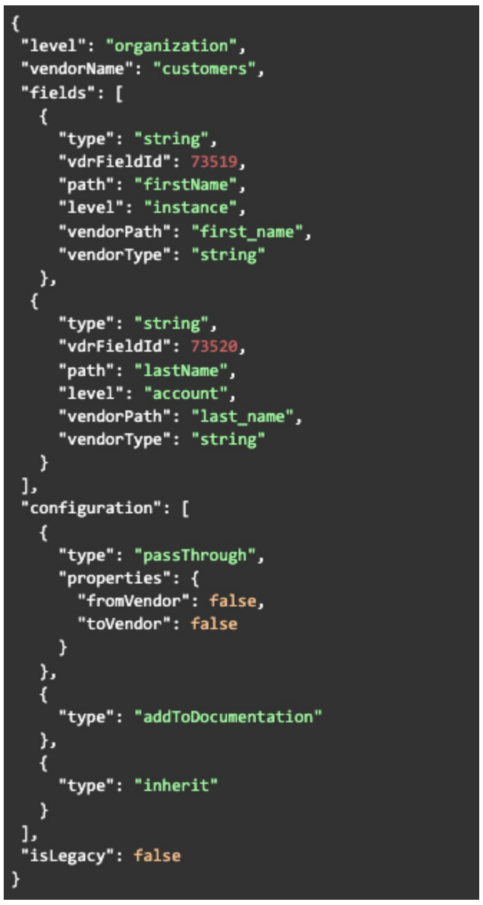
2. Call `POST` to `/accounts/objects/{objectName}/definitions` where **objectName** is the name of the common resource you want to create. You can do this via the API docs here or via a cURL command as shown below.

```
curl -X POST "https://staging.cloud-elements.com/elements/api-v2/accounts/objects/VDRDocsTes
t/definitions" -H "accept: application/json" -H "Authorization: User {user_secret}, Organiza
tion {org_secret}" -H "Content-Type: application/json" -d "{ \"fields\": [ { \"type\": \"str
ing\", \"path\": \"newName\", \"displayName\": \"\" }, { \"type\": \"string\", \"path\": \"l
ogs\", \"displayName\": \"\" } ]}"
```

3. Construct the JSON request body.
    - The id field in your response body (on the left) needs to be added as a new field for each field you created, called vdrFieldId, in your request body.
    - This is where you define your transformation, and state the path the common resource will map to on the vendor's side, i.e. the value firstName in the path field, will map to the vendorPath field first_name. This is a transformation for Sugar CRM's `/contacts` (not `/customers` like in the image below) endpoint.

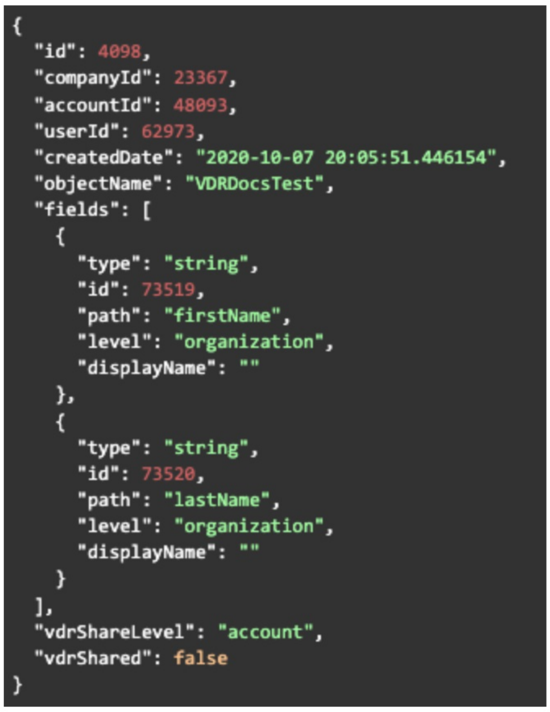
> **Note:** You can also add the **level** field to each field mapping, which will determine the level that your common resource fields will be mapped to. (See below where the firstName is mapped at the instance level while the field lastName is mapped at the account level.

Request Body



```
{
 "level": "organization",
 "vendorName": "customers",
 "fields": [
   {
     "type": "string",
     "vdrFieldId": 73519,
     "path": "firstName",
     "level": "instance",
     "vendorPath": "first_name",
     "vendorType": "string"
   },
  {
     "type": "string",
     "vdrFieldId": 73520,
     "path": "lastName",
     "level": "account",
     "vendorPath": "last_name",
     "vendorType": "string"
   }
 ],
 "configuration": [
   {
     "type": "passThrough",
     "properties": {
       "fromVendor": false,
       "toVendor": false
     }
   },
   {
     "type": "addToDocumentation"
   },
   {
     "type": "inherit"
   }
 ],
 "isLegacy": false
}
```

Response Body



```
{
  "id": 4098,
  "companyId": 23367,
  "accountId": 48093,
  "userId": 62973,
  "createdDate": "2020-10-07 20:05:51.446154",
  "objectName": "VDRDocsTest",
  "fields": [
    {
      "type": "string",
      "id": 73519,
      "path": "firstName",
      "level": "organization",
      "displayName": ""
    },
    {
      "type": "string",
      "id": 73520,
      "path": "lastName",
      "level": "organization",
      "displayName": ""
    }
  ],
  "vdrShareLevel": "account",
  "vdrShared": false
}
```

4. Call `POST` to `/accounts/objects/{objectName}/definitions` where **objectName** is the name of your common resource (as defined in step 2) and keyOrId is the connector key id . You can do this via the API docs in the UI (linked here), or via a curl command as shown below.

```
curl -X POST "https://staging.cloud-elements.com/elements/api-v2/accounts/elements/234/trans
formations/VDRDocsTest" -H "accept: application/json" -H "Authorization: User {user_secret},
Organization {org_secret}" -H "Content-Type: application/json" -d "{ \"level\": \"account\",
\"vendorName\": \"customers\", \"fields\": [ { \"type\": \"string\", \"vdrFieldId\": 73519,
\"path\": \"firstName\", \"level\": \"instance\", \"vendorPath\": \"first_name\", \"vendorTy
pe\": \"string\" }, { \"type\": \"string\", \"vdrFieldId\": 73520, \"path\": \"lastName\", \
"level\": \"account\", \"vendorPath\": \"last_name\", \"vendorType\": \"string\" } ], \"conf
iguration\": [ { \"type\": \"passThrough\", \"properties\": { \"fromVendor\": false, \"toVen
dor\": false } }, { \"type\": \"addToDocumentation\" }, { \"type\": \"inherit\" } ], \"isLeg
acy\": false}"
```

5. Your common resource and transformation is now visible in the UI and available via API with the following cURL command (note you can also POST, PATCH, and DEL if your common resource supports those request types).

```
curl -X GET "https://staging.cloud-elements.com/elements/api-v2/VDRDocTest" -H "accept: appl
ication/json" -H "Authorization: User {user_secret}, Organization {org_secret}, Element {ele
ment_instance_token}"
```

Common Resource RBAC Matrix

The matrices below cover the privileges for common resource UI and common resource sharing. You can refer to this to understand how the new UI and sharing behaves for users with different roles and privileges.

Common Object UI

Roles (Or) Privileges/Feature	Account Administrator	Default User
Create Common Object	✔	X
Delete Common Object	✔	X
Create Transformation	✔	✔

Roles (Or) Privileges/Feature	Account Administrator	Default User
Delete Transformation	✔	✔
Add/Update/Delete Account Level Fields	✔	X
Add/Update/Delete Instance Level Fields	✔	✔

Shared Common Object

Roles (Or) Privileges/ Feature	Account Administrator	Default User
Share Common Object (Account owned common Object)	X	X
Unshare Common Object (Account owned Common Object)	X	X
Delete Common Object (Account Owned Common Object)	X	X
Delete Transformation (Account owned Transformation)	X	X
Add/Update/Delete Account Level Fields	✔	X
Add/Update/Delete Instance Level Fields	✔	✔